

Rapport Flow Free



Sommaire:

Introduction	p.3
Description des fonctionnalités	p.5
Structure du Programme Flow Free	p.7
Algorithme de victoire	p.8
Conclusion	p.10

Introduction:

Flow Free est un jeu de réflexion et de logique développé par Big Duck Games LLC et publié en 2012 sur plateformes mobiles (Android, iOS). Le principe est simple : relier des paires de points de même couleur à l'aide de tracés continus appelés *flows*, tout en remplissant entièrement une grille sans que les chemins ne se croisent ni ne se superposent.

Le jeu s'inspire des casse-têtes classiques de connexion et propose des centaines de niveaux de difficulté croissante, avec des grilles allant de 5x5 à 15x15 cases, voire plus dans certaines variantes. Sa prise en main intuitive, combinée à une montée en complexité progressive, en fait un jeu à la fois accessible et stimulant pour les amateurs de logique.

Grâce à son succès, *Flow Free* a connu plusieurs déclinaisons, comme *Flow Free: Bridges*, *Hexes* ou *Warps*, et cumule aujourd'hui plus de 100 millions de téléchargements dans le monde.



Logo de Big Duck Games

Description des fonctionnalités:

1. Chargement de puzzles personnalisés

L'application permet de charger dynamiquement des grilles de jeu depuis des fichiers `.xml` placés dans le dossier `assets/puzzles/`. L'utilisateur peut sélectionner un puzzle dans une liste, ce qui rend le jeu facilement extensible et personnalisable.

2. Affichage automatique de la grille

Une grille s'affiche en fonction de la taille définie dans le puzzle. Chaque paire de points est représentée par deux cercles colorés.

3. Interaction tactile pour tracer les chemins

L'utilisateur commence un chemin en touchant un cercle coloré, puis trace le chemin en glissant le doigt vers l'autre cercle de même couleur. Les tracés sont visuellement colorés et épais, pour un rendu clair et agréable.

4. Réinitialisation d'un chemin

Si l'utilisateur souhaite recommencer un tracé, il peut cliquer de nouveau sur l'un des deux cercles d'une paire, ce qui efface automatiquement le chemin en cours et permet de le redessiner.

5. Retour en arrière

Lorsqu'un tracé est en cours, l'utilisateur peut revenir sur ses pas en glissant vers une cellule précédente du chemin. Le programme supprime alors les dernières cases du tracé.

6. Gestion des collisions et des règles de tracé

- Il est impossible de tracer un chemin à travers un point déjà utilisé ou un point appartenant à une autre paire.
- Une fois qu'une paire est correctement reliée, le chemin est verrouillé.

7. Détection automatique de la victoire

Le programme vérifie automatiquement si toutes les cases de la grille sont remplies et si toutes les paires ont été connectées. Un message de victoire s'affiche alors à l'écran.

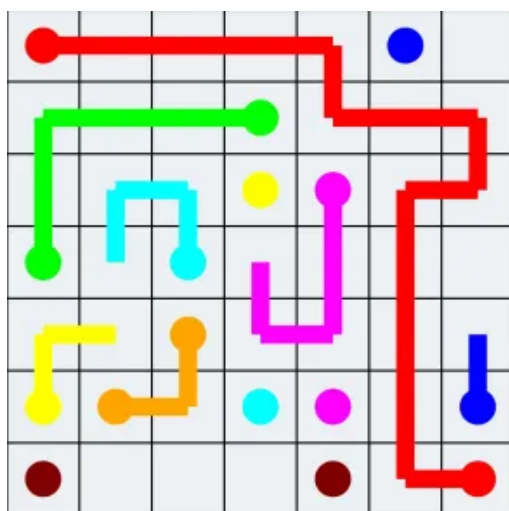
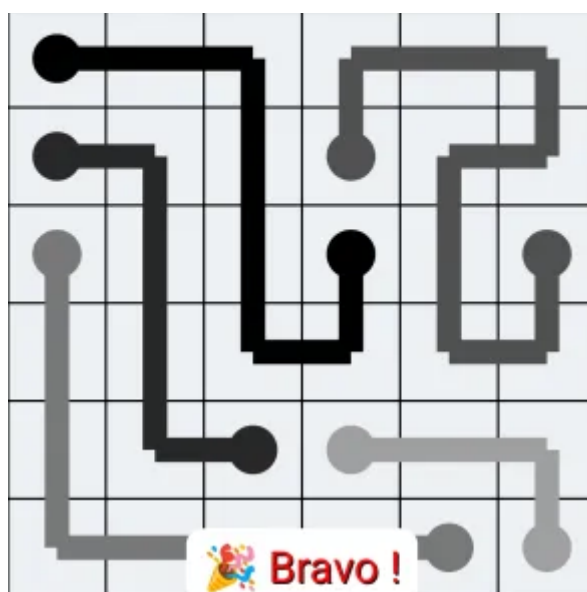


Illustration de la grille, des points et des traits (oui c'est moche sur celui là)

8. **Achromatique:** On a fait un truc pour eux



Structure du programme Flow Free

Le projet Flow Free est composé de plusieurs activités Android et de classes Java permettant de gérer l'affichage, les interactions utilisateur et la logique du jeu.

Structure globale du projet

1. Activités principales :

- **MenuActivity** : Affiche le menu principal avec le bouton "Jouer".
- **MainActivity** : Affiche la liste des puzzles disponibles et permet d'en choisir un.
- **GameActivity** : Lance le jeu avec le puzzle sélectionné.
- **SettingsActivity** : (optionnel) Paramètres du jeu.
- **RulesActivity** : Affiche les règles du jeu.

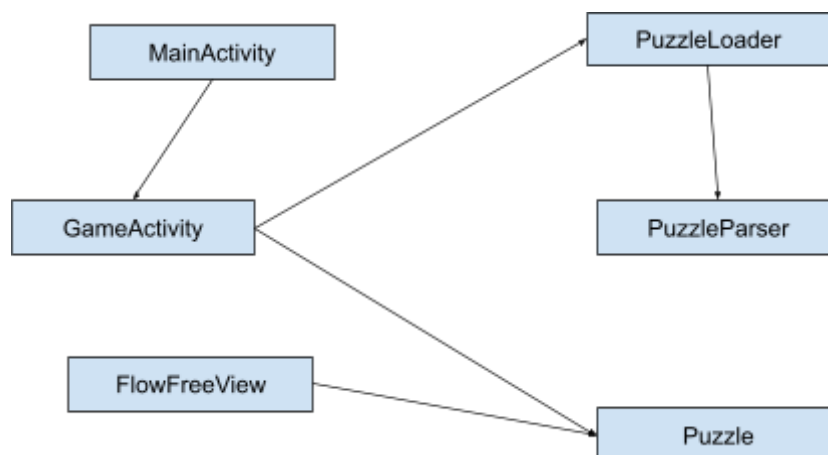
2. Vue personnalisée :

- **FlowFreeView** : Gère l'affichage de la grille, les chemins, les interactions utilisateur (touches) et la détection de victoire.

3. Modèles de données :

- **Puzzle** : Représente une grille de jeu avec sa taille et les coordonnées des paires.
- **PuzzleLoader** : Charge un puzzle depuis les fichiers.
- **PuzzleParser** : Analyse les fichiers XML et crée les objets **Puzzle** correspondants.

Graphe des relations (j'ai tenté monsieur)



Algorithme de victoire

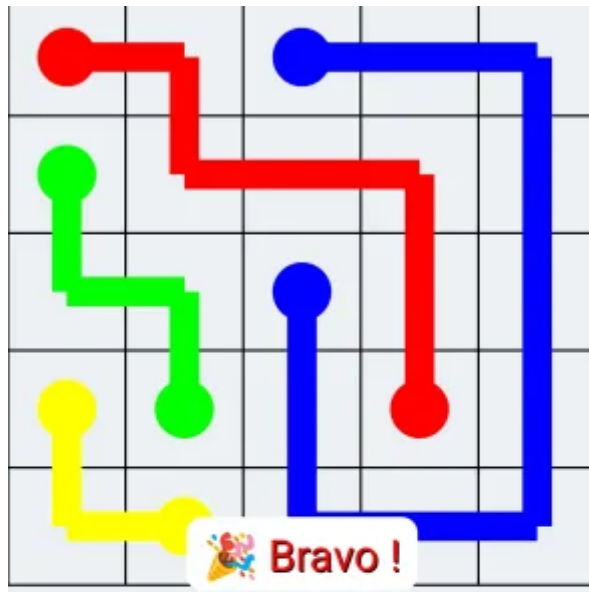
L'algorithme de victoire dans notre version du jeu Flow Free repose sur deux grandes conditions qui doivent être satisfaites pour considérer que le joueur a terminé le puzzle avec succès. Cette vérification est effectuée à la fin de chaque mouvement pertinent, notamment après la connexion d'une paire de points de même couleur.

La première condition est que la grille de jeu soit complètement remplie. Le programme parcourt l'intégralité du tableau `board`, qui représente l'état de chaque case de la grille. Chaque case contient soit un identifiant positif (correspondant à un point d'origine ou d'arrivée), soit un identifiant négatif (correspondant à un segment de chemin), soit la valeur `0` (qui signifie que la case est vide). Tant qu'au moins une case du tableau contient la valeur `0`, cela signifie qu'il reste des emplacements non couverts, donc que le puzzle est incomplet. Dans ce cas, le programme arrête immédiatement la vérification : il est inutile de vérifier les autres conditions si la grille n'est pas encore pleine.

La seconde condition est que toutes les paires de points du puzzle soient correctement reliées par un chemin. À chaque fois qu'un joueur termine un tracé entre deux points de même couleur (c'est-à-dire qu'il arrive à toucher l'extrémité d'une paire depuis l'autre), l'identifiant de cette couleur est ajouté dans un ensemble nommé `completedColors`. Lorsque l'on souhaite savoir si tous les chemins attendus ont été tracés, le programme compare simplement la taille de cet ensemble `completedColors` au nombre total de paires définies dans le puzzle. Si les deux valeurs sont égales, cela signifie que chaque paire a bien été reliée par un chemin, sans doublon.

Si ces deux conditions sont satisfaites en même temps – c'est-à-dire que toutes les cases sont remplies et toutes les paires reliées – alors la variable `win` est passée à `true`. Cela a pour effet de déclencher l'affichage d'un message de félicitations (" Bravo !"(c'était une insulte de base je balance)) directement au centre de la vue du jeu, signalant que le puzzle est terminé avec succès.

Ainsi, l'algorithme est à la fois simple, robuste et lisible, tout en assurant que le joueur a respecté les règles fondamentales du jeu : relier chaque paire et remplir toute la grille sans exception.



Conclusion:

Patrick Felix-Vimalaratnam(le Roux, c'est des ref à One Piece monsieur):

Ce projet était intéressant par le fait qu'on pouvait voir le résultat sur nos portables, découvrir la complexité d'un jeu sur mobile et ce alors que notre sujet n'est pas très poussé. De plus, de voir les différents puzzles possibles d'utilisation, m'a rappelé le projet de sudoku qui utilisait aussi cette fonctionnalité.

Marvin Aubert (l'noir):

Premier projet réussi avec vous, incroyable n'est-ce pas. C'était sympa, j'ai galéré à installer Android Studio (Patoche ça marche pas avec lui). Sinon ça va, c'était sympa, pas trop de problèmes. Quelques événements loufoques sont arrivés durant le projet comme le raid de la guêpe durant un cours ou le fond d'écran de jeu du Flow de Damien et Tom (c'était bien pire de base)

Maxime Lebreton (l'blanc):

Ce projet était vraiment intéressant. Ce que j'ai préféré, c'est le moment où toute la partie graphique était finalisée et qu'il fallait créer toutes les conditions pour que ça fonctionne. C'est vraiment ce que je préfère quand je code : trouver des solutions à des problèmes sans me soucier de l'aspect graphique.